AMENDMENTS

Please amend the present application as follows:

The following is a marked-up version of the specification with the language that is underlined ("___") being added and the language that contains strikethrough ("——") being deleted:

For paragraph [0008] beginning on page 7:

[0008] FIG. 1 is a flow diagram that depicts one example method for generating a random bit stream. According to this example method, a plurality of hardware driven numbers are accumulated (step 5). A portion of each hardware driven number is extracted (step 10). The extracted portions are combined to form a random bit stream ~~(set 15)~~(step 15). According to one alternative method, a hardware driven number comprises a plurality of bits and extraction of a portion of the number comprises extracting one or more bits from the hardware driven number. By using an extracted portion of a plurality of hardware driven ~~number~~numbers, additional entropy is tapped from the source. By varying the amount and type of extraction, varying degrees of entropy can be tapped from the source in accordance with the requirements of a consuming process, e.g. an encryption algorithm.

For paragraph [0010] beginning on page 7:

[0010] FIG. 3 depicts one alternative illustrative method for accumulating a plurality of hardware driven numbers. In order to be useful, a random bit stream must exhibit some output bit rate that is dictated by a user process, e.g. an encryption process. The encryption process, in turn, may dictate a pre-established output bit rate based on the need to secure some amount of data at a prescribed bandwidth. Hence, one example method first obtains a pre-established output bit rate (step 30). In many cases, a random bit stream generated according to this method embodiment may exhibit some level of uniformity. Where the output random bit stream does not exhibit sufficient uniformity, one example method provides for de-skewing the output random bit stream to achieve greater uniformity. Uniformity refers to a substantially equal distribution of ones and zeros in the bit stream over some period of time. A de-skewing mechanism, according to one example method, is applied to the bit stream in order to improve uniformity. One example of a de-skewing mechanism discards some number of bits in order to improve uniformity. When such a de-skewing mechanism is utilized, there is an efficiency associated with the de-skewing process. Hence, the resultant random bit stream may be of a lesser output bit rate than the random bit stream operated on by the de-skewing mechanism. Consequently, once a de-skewing mechanism is selected (step 35), a quantity of hardware driven numbers can be determined (step 40) by considering the efficiency of the selected de-skewing mechanism and the pre-established output bit rate dictated by a consuming process. The determined quantity of hardware driven numbers is then accumulated ~~(step 25)~~(step 45) and used to generate a random bit stream.

For paragraph [0011] beginning on page 8:

[0011] FIG. 4 depicts an illustrative method for combining extracted portions of hardware driven numbers to form a random bit stream. According to one alternative method, extracted portions of hardware driven numbers are concatenated (step 50). As used herein concatenation is accomplished across a plurality of hardware driven numbers from the same bit position. The concatenated result may not be uniform in distribution of ones and zeros. Accordingly, this example method further comprises de-skewing the concatenated result in order to provide a more uniform distribution of random bits (step 55). According to one alternative method, de-skewing is accomplished by state transition mapping. Hence, a state is depicted according to a numerical value represented by the concatenated result. This state then drives a mapping table. A resultant numerical value, for example a value stored in the mapping table, represents random bits that exhibit a desired level of uniformity. Application of the de-skewing method need not be limited to this one form of de-skewing. Alternative de-skewing methods may be used that rely on various types of de-skewing mechanisms, including, but not limited to~~:~~ transition mapping and exclusive-ORing of a number of bits in the concatenated result to form a single bit in a de-skewed result.

For paragraph [0015] beginning on page 9:

[0015] FIG. 6 is a block diagram of one illustrative alternative embodiment of a number receiver. According to this illustrative alternative embodiment, a number receiver comprises a time interface 150 and a buffer 170 capable of storing the timestamp 165 received by the time interface 150. The time interface 150, according to one example embodiment, retrieves a timestamp 160 from an external source (e.g. a real-time clock in a computer). The time interface 150, according to yet another alternative embodiment, obtains a new timestamp 160 on a periodic basis, the period of which is established by the number pulse 70. The time interface 150, according to yet another alternative embodiment, further is capable of generating a read signal ~~161~~ that can be used to request data from a real time clock. The read signal ~~161~~ is generated every time a number pulse 70 is received by the time interface 150. According to yet another example embodiment, the buffer 170 comprises a first-in-first-out buffer. The output of the buffer 170 constitutes a timestamp stream 180 that is used as a source of hardware driven numbers according to one method presented herein.

For paragraph [0018] beginning on page 11:

[0018] FIG. 8 is a block diagram of one example embodiment of a cycle generator.

According to this example embodiment, a cycle generator comprises a time base generator, e.g. a crystal oscillator 275, that operates at a frequency set by a crystal 280. It should be noted that any suitable time base generator may be used. The time base generator generates a time base signal 290. According to this example embodiment, the cycle generator further comprises a factor table 255. The factor table 255 generates a de-skewing factor 260. According to yet another example embodiment, the factor table 255 also generates a timing factor 270. It should be noted that the timing factor 270 is an optional factor that according to this alternative example embodiment is used as additional information by a countdown divider, as described infra. The factor table 255 generates the de-skewing factor ~~216~~260 and the optional timing factor 270 according to a bit rate indicator 250. The bit rate indicator 250 is typically received from an external source, for example a consuming process that dictates the rate at which random bits are required to support a particular encryption algorithm.

For paragraph [0022] beginning on page 12:

[0022] In operation, a counter 420 included in the de-skewing register 360 is configured according to the de-skewing factor ~~290~~260. The de-skewing factor ~~290~~260 is used to determined the quantity of valid bits that are present in the de-skewing register 360. The counter 420 operates in a periodic manner according to the time base signal 290 and uses the number pulse 70 to generate a load signal 410 and a shift signal 415. These signals control the shift register 400 so as to load a particular number of valid bits 390 from the transition mapping table 430. Once the bits are loaded into the shift register 400 according to be load signal 410, the bits are shifted out of the shift register 400 as a stream of de-skewed random bits 100. Shifting occurs according to the shift signal 415.

For paragraph [0024] beginning on page 13:

[0024] According to one example embodiment of a random number generator, instruction sequences that implement functional modules are stored in the memory 470 including a number receiver module 480, an extractor module 490 and a concatenator module 500. According to one alternative embodiment, an additional instruction sequence that implements a de-skewing module ~~500~~510 is also included in the memory 470. In this case, a portion of the memory 470, according to one alternative embodiment, is used to store a de-skewing table 530.